

**WEST**

## Freeform Search

Database: US Patents Full-Text Database  
US Pre-Grant Publication Full-Text Database  
JPO Abstracts Database  
EPO Abstracts Database  
Derwent World Patents Index  
IBM Technical Disclosure Bulletins

Term: L8 and error near4 sectio\$2

Display: 10 Documents in Display Format:  Starting with Number 1

Generate: ☐ Hit List ☒ Hit Count ☐ Side by Side ☐ Image

Search

Clear

Help

Logout

Interrupt

Main Menu

Show \$ Numbers

Edit \$ Numbers

Preferences

Cases

## Search History

DATE: Thursday, March 07, 2002 [Printable Copy](#) [Create Case](#)Set Name Query  
side by sideHit Count Set Name  
result set

DB=USPT; PLUR=YES; OP=OR

|            |  |      |            |
|------------|--|------|------------|
| <u>L10</u> | L8 and error near4 sectio\$2                               | 33   | <u>L10</u> |
| <u>L9</u>  | L8 error near4 sectio\$2                                   | 4623 | <u>L9</u>  |
| <u>L8</u>  | secondary adj4 storage and pag\$2 and primary adj4 storage | 325  | <u>L8</u>  |
| <u>L7</u>  | L6 and data adj4 record                                    | 0    | <u>L7</u>  |
| <u>L6</u>  | L5 and retrieval   | 20   | <u>L6</u>  |
| <u>L5</u>  | L4 and pag\$2  | 32   | <u>L5</u>  |
| <u>L4</u>  | L3 and control adj5 program                                | 32   | <u>L4</u>  |
| <u>L3</u>  | L2 and secondary near4 storage                             | 36   | <u>L3</u>  |
| <u>L2</u>  | L1 and primary near4 storage                               | 53   | <u>L2</u>  |
| <u>L1</u>  | storage and sectio\$2 near4 erro\$2                        | 1981 | <u>L1</u>  |

END OF SEARCH HISTORY

**WEST**[Help](#)[Logout](#)[Interrupt](#)[Main Menu](#)[Search Form](#)[Posting Counts](#)[Show S Numbers](#)[Edit S Numbers](#)[Preferences](#)[Cases](#)**Search Results -**

| Terms                    | Documents |
|--------------------------|-----------|
| L5 and (cache or memory) | 10        |

**Database:**

US Patents Full-Text Database  
 US Pre-Grant Publication Full-Text Database  
 JPO Abstracts Database  
 EPO Abstracts Database  
 Derwent World Patents Index  
 IBM Technical Disclosure Bulletins

**Search:**

[Refine Search](#)[Recall Text](#)[Clear](#)**Search History**
**DATE: Thursday, March 07, 2002**
[Printable Copy](#)
[Create Case](#)
**Set Name**  
 side by side
**Query**
**Hit Count**
**Set Name**  
 result set

*DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR*

|           |                                     |      |           |
|-----------|-------------------------------------|------|-----------|
| <u>L7</u> | L5 and (cache or memory)            | 10   | <u>L7</u> |
| <u>L6</u> | L5 and cache                        | 0    | <u>L6</u> |
| <u>L5</u> | L4 and stor\$                       | 11   | <u>L5</u> |
| <u>L4</u> | L2 and data near record             | 12   | <u>L4</u> |
| <u>L3</u> | L2 and data near record near stor\$ | 0    | <u>L3</u> |
| <u>L2</u> | L1 and client adj application       | 84   | <u>L2</u> |
| <u>L1</u> | database adj manager                | 1372 | <u>L1</u> |

END OF SEARCH HISTORY

**WEST**☐  

L7: Entry 7 of 10

File: USPT

Feb 16, 1999

US-PAT-NO: 5873086

DOCUMENT-IDENTIFIER: US 5873086 A

TITLE: Communications control apparatus and client/server computer system

DATE-ISSUED: February 16, 1999

## INVENTOR-INFORMATION:

| NAME                | CITY     | STATE | ZIP CODE | COUNTRY |
|---------------------|----------|-------|----------|---------|
| Fujii; Yasushi      | Kawasaki |       |          | JPX     |
| Imura; Hideo        | Kobe     |       |          | JPX     |
| Hashizume; Yasuhiko | Kawasaki |       |          | JPX     |

## ASSIGNEE-INFORMATION:

| NAME            | CITY     | STATE | ZIP CODE | COUNTRY | TYPE CODE |
|-----------------|----------|-------|----------|---------|-----------|
| Fujitsu Limited | Kawasaki |       |          | JPX     | 03        |

APPL-NO: 8/ 968477 [PALM]

DATE FILED: November 12, 1997

## PARENT-CASE:

This application is a continuation of application Ser. No. 08/433,732, filed May 3, 1995, now abandoned.

## FOREIGN-APPL-PRIORITY-DATA:

| COUNTRY | APPL-NO  | APPL-DATE    |
|---------|----------|--------------|
| JP      | 6-121957 | May 10, 1994 |

INT-CL: [6] G06 F 17/30

US-CL-ISSUED: 707/10

US-CL-CURRENT: 707/10

FIELD-OF-SEARCH: 707/10

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

**WEST**

Generate Collection

Print

L7: Entry 7 of 10

File: USPT

Feb 16, 1999

DOCUMENT-IDENTIFIER: US 5873086 A

TITLE: Communications control apparatus and client/server computer system

Abstract Paragraph Left (1):

A communication control apparatus controls the transmission of data between client and server applications in a client/server system. The alteration to the system of a server machine does not affect the client application by temporarily storing the data in an internal database.

Abstract Paragraph Left (2):

The communication control apparatus contains a queue-structure entry table or response table as the database storing data notified of by the client application or the response data by the server application. It also stores the table defining information which defines the format of each table. The data transmitted between the client application and the server application is queued and retrieved on each of the tables.

Brief Summary Paragraph Right (2):

The present invention relates to a client/server computer system. In particular, the present invention relates to a communication control apparatus for controlling message exchange in a client/server computer system having a front-end process application (client application) that is disposed in a client machine and a back-end process application (server application) that is disposed in a server machine and a client/server system thereof.

Brief Summary Paragraph Right (6):

In the client/server type database system, database access types of a database server (computer) machine (referred to as a database server) by a client application mounted on a client machine can be categorized as RDA (Remote Database Access) and RPC (Remote Procedure Call).

Brief Summary Paragraph Right (7):

In the RDA type of database access, a client application accesses a database using the RDA protocol established by the ISO (International Organization for Standardization). In this case, the client application directly sends a structured query language (SQL) statement that is received through a network to a DBMS (Data Base Management System). The results retrieved from the DBMS is sent to the client application by middle-ware (referred to as the RDA server). Thus, the designer of the client application should know the construction of the database on the database server.

Brief Summary Paragraph Right (9):

First, a problem of a C/S database system (client/server type database system) in which a client application accesses the database server corresponding to the RDA will be described.

Brief Summary Paragraph Right (10):

In this type, the developer of the client application should know the construction of the database on the database server. Thus, the database manager who designs the database and the designer who designs the client application should work in liaison with each other.

Brief Summary Paragraph Right (11):

In addition, the content of the client application mutually affects the construction of the database. Thus, when the manager of the database changes the construction of

the database, the manager should consider the influence of the change against the existing client application. Consequently, the database manager should be familiar with the content of the client application.

Brief Summary Paragraph Right (12):

In such a construction, when the construction of the database is changed (for example, the database is divided or data position is changed), the client application should be inevitably modified.

Brief Summary Paragraph Right (13):

FIGS. 1A and 1B are an example of a change of the construction of the database. In this case, since the number of items of a table 2 (TABLE1) of a database DB1 of a database server 1 is increased due to an increase of transactions, the access performance is deteriorated. To improve the access performance, as shown in FIG. 1B, the table 2 (TABLE1) is divided into a table 3 (TABLE1-1) and a table 4 (TABLE1-2). Thus, a part of a client application 7 on a client machine 6 is corrected from a content (INSERT TABLE1) shown in FIG. 1A to a content (INSERT TABLE1-1, INSERT TABLE1-2) shown in FIG. 1B. Thus, the correcting work results in an increase of the cost. In addition, the client machine cannot be temporarily used.

Brief Summary Paragraph Right (14):

In addition, the manager of the database server should preform a modification work such as dispersion of load of the database and change of data management unit when the client application is changed. Thus, the period for the modification work available for the manager of the database server is restricted.

Brief Summary Paragraph Right (16):

As shown in FIG. 2, when an order process is performed in the RDA, a client application 11 requests a database server 20 to perform a process such as a customer check, an inventory check, an inventory update, or an order registration with an SQL statement. Each SQL statement is issued after a response to the former SQL statement has been received from the database server 20. Thus, while the order process is being executed, data is repeatedly exchanged between the client machine 10 and the database server 20. Consequently, the traffic amount of the network may increase, thereby decreasing the process performance of the system.

Brief Summary Paragraph Right (17):

In the RDA, a request of an SQL statement from the client application is immediately sent to the DBMS through the RDA server. Thus, even if the system manager establishes the following rules (1) to (4) as database operation rules, he cannot substantially manage/control the operation of the system.

Brief Summary Paragraph Right (22):

Other than the process request method using the SQL statement for accessing the client application corresponding to the RDA, a so-called stored procedure call method is available. In the stored procedure call method, a sequence of transaction processes (procedures) are pre-registered in the database server. By calling a procedure, a predetermined database process is performed. However, in the present situation, since the client application should generate a job logic by accessing the database with the SQL statement and using the stored procedure call, the job logic becomes complicated, thereby preventing the productivity from being improved.

Brief Summary Paragraph Right (24):

Next, a so-called C/S application type database system will be described. In this database system, applications are categorized as a client application that performs input/output processes for the user interface and a server application that executes job logic for database access. The client application and the server application are mounted on a client machine and a database server, respectively. The client application calls the server application corresponding to the RPC so as to execute the job process. This C/S application type database system has the following problems.

Brief Summary Paragraph Right (26):

As the above-described process type, real communication and delayed communication are known. In the real communication, when the server application has completed a process requested by the client application, the client application is informed of the completion of the process.

Brief Summary Paragraph Right (27):

On the other hand, in the delayed communication, when a message (data) received from the client application is stored without need to wait until the process of the server application has been completed, the client application is informed of the completion of the process.

Brief Summary Paragraph Right (28):

Thus, when the client application performs the collecting type process that collects data for the server application on a real time communication basis, if the process time of the server application becomes very long, the reply time for the client application is delayed. In this case, the collecting type process should be changed from "real communication" to "delayed communication". Thus, the development of both the applications has a close relationship to each other. Consequently, it is very difficult to independently develop them. In addition, since the change of the process type affects the logic of the application, both the applications should be changed in many situations, thereby deteriorating the productivity.

Brief Summary Paragraph Right (29):

Since the client application performs a process for the user interface, the attribute (type) of data to be processed should be suitable for a screen process. On the other hand, since the server application accesses the database and edits the data thereof, the application should be preferably processed with attributes suitable for such processes. Thus, when data is exchanged between the client application and the server application, the data attribute changing process is required on either side.

Brief Summary Paragraph Right (31):

As shown in FIG. 4A, the operation test for an application that is newly developed cannot be performed unless both the client application 31 and the server application 32 are provided. For a test in the RPC, as shown in FIGS. 4B and 4C, a driver 36 and a stub 37 should be provided.

Brief Summary Paragraph Right (32):

In other words, when only the server application 32 is tested, a driver that issues a request to the server application 32 instead of the client application 31 should be provided (FIG. 4B). On the other hand, when only the client application 31 is tested, the stub 37 that receives a request from the client application 31 instead of the server application 32 should be provided (FIG. 4C).

Brief Summary Paragraph Right (34):

In addition, since the client application and the server application are in liaison with each other through a TP monitor, they should be generated corresponding to an API (Application Programming Interface) of the TP monitor. Thus, in this situation, it is impossible to construct the C/S application type database system corresponding to the RPC using commercial package software (such as spreadsheet software, database, or access software) mounted on a PC (personal computer) as a client application.

Brief Summary Paragraph Right (35):

In addition, in the existing C/S application type database system, the server application designated by the client application corresponding to the RPC is operated only in the same executing environment on the database server. In other words, the same server application can be used in one job process. Thus, when the existing server application is used for a new job process, the name should be changed. Consequently, various jobs involving the name change are required.

Brief Summary Paragraph Right (36):

In other words, as shown in FIG. 5, in a system that is operating, a client application 41 calls a server application A. Thereafter, so as to extend the job, using the server application A, a new client application 42 is developed (1). Since the new job process becomes another executing environment, the server application A should be managed by a server application A' with another name (2). However, in this case, when the server application A is corrected due to an occurrence of a trouble, not only the server application A, but the server application A' should be corrected (3). In other words, the same server application A should be dually managed, thereby increasing the maintenance work.

Brief Summary Paragraph Right (37):

The object of the present is to provide a communications control device and a client/server system equipped with the communications control device which enables

the development and alteration of client and server applications to be performed independently in a client/server system and enables the structure of a database to be altered without modifying the client application.

Brief Summary Paragraph Right (38):

The communications control device according to the present invention comprises a first database storing data transmitted between a client application and a server application and a control portion for controlling the data communications between the client application and the server application by temporarily storing in the first database the data received from the client or server application.

Brief Summary Paragraph Right (39):

According to the present invention, temporarily storing data in the first database in data communications between the client and server applications enables the structure of the database to be successfully changed in the server application. Therefore, changing the structure of the database in server equipment does not require a change of the client application.

Brief Summary Paragraph Right (41):

Thus, the client application generates data to be transmitted taking only the published service information into account without considering the structure of the database in the server equipment.

Drawing Description Paragraph Right (25):

FIG. 25 is a flowchart for explaining an item attribute converting process in the case that a table storing request is received from a client application;

Drawing Description Paragraph Right (28):

FIG. 28 is a flowchart for explaining an operation in the case that a request for extracting data queued in a table is received from a client application;

Detailed Description Paragraph Right (2):

The present invention is a client/server system that executes a job process by a client application 52 mounted on a client machine 50 and a server application 62 mounted on a server machine 60. The client/server system is a communication control apparatus that controls information exchange between the client application 52 and the server application 60.

Detailed Description Paragraph Right (3):

The client/server system shown in FIG. 6 includes a communication database (communication DB70) between a client application 52 in a client machine (computer) 50 and a server application 62 in a server machine (computer) 60. The communication DB 70 stores public information open (provided) to a client and information being input from the client as a database and retrieves information to be sent to the server application from the database (not shown). The server application 62 receives the information retrieved from the database and accesses the database 63.

Detailed Description Paragraph Right (10):

The first queue control portion 72 informs a requesting client application 52 of the completion of a requested process for example when the data has been queued to the first input table 73.

Detailed Description Paragraph Right (11):

The first queue control portion 72 does not queue data requested by the client application 52 to the first input table 73 for example when a service of public service information that defines the format of the first input table 73 is caused to be closed.

Detailed Description Paragraph Right (16):

A first attribute converting portion 83 converts an attribute of each item of data received from the client application 52 into an attribute defined in the public service information and outputs the converted attribute to the first queue control portion 72.

Detailed Description Paragraph Right (18):

A second informing portion 78 informs the client application 52 of data to be sent to the client application 52 queued in the reply table 76 corresponding to a request from the client application 52.

Detailed Description Paragraph Right (19):

The second informing portion 78 references attribute information defined in the public service definition information and converts an attribute of each item of data retrieved from the reply table 76 to an attribute required by the client application 52 for example when the data retrieved from the reply table 76 is sent to the client application 52.

Detailed Description Paragraph Right (26):

A third informing portion 81 informs the client application 52 of data to be sent to the client application 52 queued in the second input table 79 corresponding to a request from the client application 52.

Detailed Description Paragraph Right (27):

The third informing portion 81 references attribute information defined in the public service definition information and converts an attribute of each item of data retrieved from the second input table 79 into an attribute requested by the client application 52 for example when the data retrieved from the second input table 79 is sent to the client application 52.

Detailed Description Paragraph Right (31):

In this construction, first security information for a client with an access right to the communication database 70 is registered at the communication database 70. In this case, the registering portion 82 may register the first security information. This construction may further comprise a first security check portion (not shown) that references the first security information and determines whether or not to allow a client to issue a connection request to the communication database 70 through the client application 52.

Detailed Description Paragraph Right (32):

In addition to the above-described construction, second security information for a user with an access right to the public service defined in the public service information is registered at the communication database 70. The registering portion 82 may register the second security information. A second security check portion references the second security information to determine whether or not to allow a client to access a table defined by the public service information through the client application 52.

Detailed Description Paragraph Right (36):

Since a communication database 70 is disposed as a communication control apparatus between a server application and a client application, the change of the construction of the database 63 on the server machine 60 can be absorbed by the server application 62. Thus, even if the construction of the database 63 of the server machine 60 is changed, it is not necessary to correct the client application 52. In addition, the communication control apparatus of the present invention is constructed as a database, and the client application can access the communication control apparatus as a database. The client application can be generated by considering only public information open (provided) to the client. In other words, it is not necessary to consider the construction of the database on the server machine 60.

Detailed Description Paragraph Right (37):

When a client application 52 sends data (message) to a server application 62, the data is assembled in a record format of a first input table 73 defined with public service information registered by a public service information storing portion 71. A server machine is requested to store the data in the first input table 73.

Detailed Description Paragraph Right (42):

As described above, the client application 52 sends data with the same format as a record format of the first input table 73 defined by the public service information to the server application 62. The server application 62 performs a data process such as database accessing with the received data.

Detailed Description Paragraph Right (43):

In this case, the record format of the first input table 73 defined by the public service information functions as an API (Application Programming Interface) for the client application 52 and the server application 62. Thus, the client application 52 and the server application 62 can be independently generated corresponding to the API without need to consider the other application.



Detailed Description Paragraph Right (44):

Consequently, even if the construction of the database is changed, it is not necessary to correct the client application 52 rather than the server application 62. Thus, the database manager can change the construction of the database without need to consider the client application 52. A table open to the user and a table constructed on the database can be independently designed.

Detailed Description Paragraph Right (45):

The first queue control portion 72 informs a requesting client application 52 of the completion of a requested process for example when the data has been queued to the first input table 73. Thus, the delayed communication can be performed.

Detailed Description Paragraph Right (46):

The first queue control portion 72 does not queue data requested by the client application 52 to the first input table 73 for example when a service of public service information that defines the format of the first input table 73 is caused to be closed. Alternatively, the server application activation request portion 74 does not issue an activation request for the server application 62 for example when data to be sent to the server application 62 that is stopped is queued in the first input table 73. By such control, the manager can prohibit the client/server database system from being accessed for a large amount of data, prohibit a client from leaving his seat while accessing the database, reserve database maintenance time, prohibit data from being input in an overload situation of the database server, and control priorities for various job processes (services).

Detailed Description Paragraph Right (47):

A first attribute converting portion 83 converts an attribute of each item of data received from the client application 52 into an attribute defined by the public service information and outputs the data with the converted attribute to a first queue control portion 72. The first queue control portion 72 queues the data to the first input table 73.

Detailed Description Paragraph Right (49):

Thus, in each job process, the client application 52 and the server application 62 process each item of data corresponding to an attribute (format) suitable to their data process without need to perform a data attribute converting process.

Detailed Description Paragraph Right (51):

A second informing portion 78 retrieves data to be sent to the client application 52 and informs the client application 52 of the data when the client application 52 issues a data receive request.

Detailed Description Paragraph Right (52):

Thus, a reply type (inquiry type) job process can be constructed between the client application 52 and the server application 62.

Detailed Description Paragraph Right (54):

A second informing portion 78 references attribute information defined in the public service definition information and converts an attribute of each item of data retrieved from the reply table 76 to an attribute required by the client application 52 when the data retrieved from the reply table 76 is sent to the client application 52.

Detailed Description Paragraph Right (55):

Thus, in the reply type job process, the client application 52 and the server application 62 can process data corresponding to attributes suitable for their process formats without need to perform a data attribute converting process in each application.

Detailed Description Paragraph Right (56):

A third queue control portion 80 queues data stored in the second input table defined by the public service information registered in the public service information storing portion 71 and received from the server application 62 to the second input table 79. A third informing portion 81 informs the client application 52 of data to be sent to the client application 52 queued in the second input table 79 corresponding to a request from the client application 52.

Detailed Description Paragraph Right (57):

Thus, a delivering type process that sends predetermined data and message from the

server application 62 to the client application 52 can be performed.

Detailed Description Paragraph Right (59):

Thereafter, a third informing portion 81 references attribute information defined in the public service definition information and converts an attribute of each item of data retrieved from the second input table 79 into an attribute requested by the client application 52 when the data retrieved from the second input table 79 is sent to the client application 52.

Detailed Description Paragraph Right (60):

Thus, in a delivering type process that sends data from the server application 62 to the client application 52, the server application 62 and the client application 52 can process data corresponding to attribute suitable to their process formats without need to perform a data attribute converting process.

Detailed Description Paragraph Right (61):

A registering portion 82 registers the public service information stored in the public service information storing portion 71. Thus, not only when a system design work is performed after the system has been installed, but while the system is being operated, a new job process can be added. In other words, by registering public service information suitable for a new job process to the storing portion 71 and generating the client application 52 and the server application 62 corresponding to the public service information, a new job process can be constructed.

Detailed Description Paragraph Right (63):

Thus, the server application 62 accesses a plurality of tables of a database corresponding to data received from the client application 52 and performs various data processes (retrieving, correcting, and editing data). Thus, a function similar to the RPC (Remote Procedure Call) can be accomplished.

Detailed Description Paragraph Right (64):

When a client issues a connection request to the communication database 70 through a client application 52, a first security check portion references the first security information of the client with an access right to the communication database 70 and determines whether or not to allow the client to access the communication database 70.

Detailed Description Paragraph Right (66):

When a client issues an access request to a table defined by the public service information through the client application 52, a second security check portion references the second security information of the client with an access right to the table defined by the public service information registered in the public service information storing portion 71 and determines whether or not to allow the client to access the table.

Detailed Description Paragraph Right (70):

In FIG. 8, a client machine 100 and a database server 200 are connected through an LAN (Local Area Network) or WAN (Wide Area Network). A console 600 is connected to the data server 200. This system is a client/server application type database system. The client machine 100 mounts a client application 110. The database server 200 mounts a server application 210. The client application 110 and the server application 210 jointly executes a job process.

Detailed Description Paragraph Right (71):

The database server 200 contains a communication DB control system 220. The communication DB control system 220 is disposed between the client application 110 and the server application 210 so as to provide an API (Application Program Interface) therebetween. The communication DB control system 220 includes a communication DB 221 that provides the user (client) for various public services. The client application 110 and the server application 210 are generated corresponding to the public services.

Detailed Description Paragraph Right (74):

A communication procedure control portion 201 sends and receives a message (data) to and from the client application 110 through a network corresponding to a communication protocol such as a TCP/IP protocol. When the communication procedure control portion 201 receives an SQL statement (INSERT) from the client application 110, the communication procedure control portion 201 performs a user qualifying process that determines whether or not the user who has executed the client

application 110 can access the database server 200.

Detailed Description Paragraph Right (78):

When the client application 110 requests the user qualifying portion 202 to execute an SQL statement, this portion 202 also performs the determining process for the user qualification.

Detailed Description Paragraph Right (81):

The communication DB (communication database) 221 is a special database that handles tentative data that is exchanged between the client application 110 and the server application 210. The communication DB 221 provides two data management languages (DML) that are "INSERT" and "SELECT" as an API for the client application 110 and the server application 210.

Detailed Description Paragraph Right (86):

The communication methods of various data process types are categorized as a real communication and a delayed communication. In FIGS. 10A to 10D, an input table 351 functions as a queue (request queue) of data that the client application 110 informs the server application 210 with an "INSERT" statement. On the other hand, a reply table 352 functions as a queue (response queue) of which the server application 210 informs the client application with the "INSERT" statement.

Detailed Description Paragraph Right (88):

In the reply type (real communication) data process, the client application 110 assembles request data and requests the database server 200 for the data with the "INSERT" statement (1). This data is queued to the input table 351. When the server application 210 issues the "SELECT" statement, the server application 210 reads the queued data (2). The server application 210 accesses the database system 240 corresponding to the data being read so as to perform check and predetermined processes. Thereafter, the server application 210 queues the processed results to the reply table 352 with the "INSERT" statement (3). When the server application 210 issues a RETURN command, the client application 110 is informed of the completion of the processes (4). After the client application 110 receives the RETURN command, it issues the "SELECT" statement and receives the process results queued in the reply table 352 (5). When the server application 210 detects an error in the data being checked, it informs the client application 110 of the error. As described above, in the reply (real communication) data process, the client application 110 and the server application 120 interactively execute a job process.

Detailed Description Paragraph Right (89):

FIG. 10B is a schematic diagram for explaining a collecting type (real communication) data process. In this process, only the input table 351 is registered in the service definition information 301. In this process, the client application 110 forcibly send data to the server application 210. In other words, the client application 110 queues data to be sent to the server application 210 to the input table 351 with the "INSERT" statement (1). The server application 210 issues the "SELECT" statement and receives the data (2). In the same manner as shown in FIG. 10A, the server application 210 accesses the database system 240 so as to perform check and predetermined processes. After the server application 210 has completed the processes, it informs the client application 110 of the process results (normal or error) with the RETURN command (3). After the client application 110 has completed the processes, in the same manner as shown in FIG. 10A, it sends the next data to the server application 210. Thus, since data that has not processed does not remains in the database server 200, other jobs are not adversely affected.

Detailed Description Paragraph Right (90):

FIG. 10C is a schematic diagram for explaining a collecting type (delayed communication) data process. This process is different from the above-described collecting type (real communication) data process in the following points. When the client application 110 sends data (message) to the communication DB control system 220 with the "INSERT" statement (1), the communication DB control system 220 queues the data (message) to the input table 351. When the data is assured, the client application 110 is informed of the completion of the process corresponding to the "INSERT" statement (2). In this case, the message data of the client application 110 is stacked in a communication data assuring file stored in a nonvolatile storage medium of a message assuring control portion 228 (that will be described later). Thus, the problem of the residual data in the database server 200 is solved.

Detailed Description Paragraph Right (92):

As described above, in the collecting type (real communication) data process, unless the server application 210 operates, the client application 110 cannot execute the next data process. On the other hand, in the collecting type (delayed communication) data process, regardless of the operation of the server application 210, the client application 110 can immediately execute the next process. Thus, the collecting type (real communication) process and the collecting type (delayed communication) process can be used depending on whether the data to be collected is immediate or not. In other words, when collected data can be immediately used for another process, the real communication is used. When the collected data is processed in a predetermined time range as in a shipping instruction, the delayed communication is used.

Detailed Description Paragraph Right (93):

FIG. 10D is a schematic diagram for explaining a delivering type (delayed communication) data process. In this process, the server application 210 queues data to be sent to the client application 110 to an input table 353 with the "INSERT" statement (1). When the communication DB control system 220 stacks the message data in a communication data assuring file of the message assuring control portion 228, it sends a completion message to the server application 210 (2). When the client application 110 requires data queued in the input table 353, it issues the "SELECT" statement and receives the data (6).

Detailed Description Paragraph Right (94):

In this data process, when the client application 110 requires data processed by the server application 210, it receives the data. For example, the database server 200 processes data of another job on batch basis. The process results are sent to the client machine 100.

Detailed Description Paragraph Right (95):

Table definition information 303I and 303R define the formats of data (record) processed corresponding to the data process type defined by local service definition information 301.

Detailed Description Paragraph Right (97):

The above-described public service information 300 is open to the user as an API for constructing a job process (namely information for generating the client application 110). In other words, the public service information 300 is a public service for which the database server 200 can provide the user so as to construct a job process.

Detailed Description Paragraph Right (99):

In addition, corresponding to the content of the public service information 300, the server application 210 is developed. In other words, by registering the public service information 300 to the communication DB 221, when the system is designed at the initial stage and after the system has been introduced, a job process can be added and/or changed. In other words, when the system is initially designed or a job process is added or changed after the system has been introduced, the public service information 300 that defines the content of the job process should be registered to the communication DB 221. When the job process is developed or added/changed, the development works of the client application 110 and the server application 210 can be independently performed.

Detailed Description Paragraph Right (101):

An SQL analyzing control portion 222C analyzes and determines an SQL received from the client application 110 through the communication procedure control portion 201. In other words, the SQL analyzing control portion 222C analyzes the SQL statement and determines whether or not the user has an access right to access the resource (table) of the communication DB 221. When the user has a proper access right, the registered data of the SQL statement to the input table of the communication DB 221 is output to a code converting/attribute converting portion 223C. The access right of the user to the table is determined by a resource use qualifying portion 224.

Detailed Description Paragraph Right (102):

The resource use qualifying portion 224 receives a user identifier of an executer of the client application 110 that has sent the SQL statement through the SQL statement analyzing control portion 222C and the information of the table of the communication DB 221 that the SQL statement accesses and determines whether or not the user has the access right to the table corresponding to security information (that will be described later) of the table registered in the communication DB 221. The resource use qualifying portion 224 sends the determined result to the SQL analyzing control

portion 222C.

Detailed Description Paragraph Right (106):

The message queue control portion 225 performs a waiting control (queue control) that waits for a message data issued from the client application 110 and the server application 210. When the waiting control is performed, the input tables 351 and the 353 and the reply table 352 shown in FIGS. 10A to 10D are used. The tables 351 to 353 are generated by the message queue control portion 225 in memory.

Detailed Description Paragraph Right (108):

The message queue control portion 225 reads input table definition information 303I that defines the format of the input table 351 registered in the communication DB 221 from the public service information 300 and generates a queue node 410 in the memory 400. The queue node 410 is composed of a queue start pointer 411, a queue last pointer 412, and table information 413. The content of the table information 413 is the same as that of the input definition information 303I. The queue node 410 manages a linking-list-structured queue generated in the communication buffer 226.

Detailed Description Paragraph Right (109):

The queue is a link-structured list in which a cell 500 composed of a chain portion 501 and a data portion 502 is connected by a pointer in the chain portion 501. The data portion 502 stores message data supplied from the client application 110 to the server application 210. The message queue control portion 225 controls the queue start pointer 411 and queue last pointer 412 of the queue node 410 and the chain portion 501 of the cell 500 so as to perform the queuing control of the message data that is input to the input table 351 or retrieved from the input table 351.

Detailed Description Paragraph Right (112):

The message assuring control portion 228 shown in FIG. 8 is provided for preventing message data from disappearing due to various error states (such as power off) of the database server 200 while message data is being sent from the database server 200 to the server application 210 or the client application 110 in the delayed communication data process (namely, collecting type delayed communication or delivering type delayed communication). The message data is stored in a communication data assuring file (not shown) of a non-volatile storage medium.

Detailed Description Paragraph Right (119):

The SQL analyzing control portion 222S receives the data process result (namely, message data to be sent to the client application 110) from the server application 210 that performs the reply type (real communication) data process with the SQL statement (INSERT). The SQL analyzing control portion 222S sends the message data to the code converting/attribute converting portion 223S so as to request it to convert the attribute.

Detailed Description Paragraph Right (123):

When the process has been completed in the real communication, the server application 210 executes the RETURN command so as to request the message reply control portion 230 to send the process request to the client application 110.

Detailed Description Paragraph Right (124):

The message reply control portion 230 notifies the client application 110 of the execution completion of the SQL statement for the database server 200 through the communication procedure control portion 201.

Detailed Description Paragraph Right (127):

FIG. 13 is a schematic diagram showing a command for the communication DB control system 220 and a data flow therethrough in the case of the reply type data process. In addition, FIG. 13 shows that communication data between the server application 210 and another client application is stored as functioning as queue in the input table 351 and the reply table 352.

Detailed Description Paragraph Right (128):

Next, the flows of the four data process types shown in FIGS. 10A to 10D that are performed between the client application 110 and the server application 210 will be described corresponding to the embodiment shown in FIG. 8.

Detailed Description Paragraph Right (129):

<Client application 110.fwdarw.communication DB control system 220>

Detailed Description Paragraph Right (130):

(1) When the client application 110 executes the SQL statement (INSERT), the communication procedure control portion 201 receives the SQL statement and qualifies the user through the user qualifying portion 202. When the communication procedure control portion 201 has qualified the access right to the communication DB 221, the portion 201 sends the SQL statement to the SQL analyzing control portion 222C.

Detailed Description Paragraph Right (138):

(7) The server application 210 issues the SQL statement (SELECT) to the SQL analyzing control portion 222S so as to receive message data from the input table 351 through the client application 110.

Detailed Description Paragraph Right (145):

(13) When the code converting/attribute converting portion 223S receives a request from the SQL analyzing control portion 222S, the portion 223 converts an attribute of the process result (message data to be sent to the client application 110) corresponding to the table attribute of the reply data 352 registered in the communication DB 221.

Detailed Description Paragraph Right (149):

(16) When the process has been completed, the server application 210 requests the message reply control portion 230 to inform the client application 110 of the completion of the execution of the SQL statement.

Detailed Description Paragraph Right (150):

(17) The message reply control portion 230 informs the client application 110 of the completion of the execution of the SQL statement (INSERT) through the communication procedure control portion 201.

Detailed Description Paragraph Right (151):

<Client application 110.fwdarw.communication DB control system 220>

Detailed Description Paragraph Right (152):

(18) The client application 110 issues an SQL statement (SELECT) to the database server 200 so as to receive reply data (message data) from the server application 210.

Detailed Description Paragraph Right (155):

(21) The code converting/attribute converting portion 223C receives a request from the SQL analyzing control portion 222C and converts the reply data into the data attribute for the client application 110.

Detailed Description Paragraph Right (156):

(22) The SQL analyzing control portion 222C informs the client application 110 of the completion of the SQL statement through the communication procedure control portion 201 along with the converted reply data.

Detailed Description Paragraph Right (158):

<Client application 110.fwdarw.communication DB control system 220>

Detailed Description Paragraph Right (167):

<Client application 110.fwdarw.Communication DB control system 220>

Detailed Description Paragraph Right (169):

(4) The message queue control portion 225 requests the message assuring control portion 228 to cause the communication data assuring file to store and hold the message data to be sent to the server application 210 queued in the input table 351. When the message queue control portion 225 receives a message that represents the completion of the process from the message assuring control portion 228, the portion 225 requests the message reply control portion 230 to issue a message that represents the completion of the execution of the SQL statement (INSERT) to be sent to the client application 110. When the message reply control portion 230 receives this message, it informs the client application 110 of the completion of the execution of the SQL statement (INSERT) through the communication procedure control portion 201.

Detailed Description Paragraph Right (179):

FIG. 14 shows a job process that generates a shipment slip in the case that a customer orders a product (such as a 29-inch color TV set). A client application 110

and a server application 210 that jointly perform such a job process are generated corresponding to public service information registered in a communication DB 221 of a communication DB control system 220 of the database server 200.

Detailed Description Paragraph Right (184):

The client application 110 generated corresponding to the public service information 300 of the public service A is composed of the following portions (1) to (4).

Detailed Description Paragraph Right (189):

Since the data format of the replay result has been defined in the reply table definition table 303R of the public service A in the communication DB 221, the editing process is performed corresponding to the definition. Thus, the client application 110 is generated by referencing the public service information 300 of the public service A registered in the communication DB 221 of the database server 200.

Detailed Description Paragraph Right (190):

On the other hand, the server application 210 that receives a request from the client application 110 through the communication DB control system 220 and that processes the request is generated corresponding to the content of the public service information 300 of the public service A.

Detailed Description Paragraph Right (192):

(1) A process that requests the communication DB control system 220 to send the message data (input data of the client application 110) queued in the input table 351 of the public service A and that receives the message data from the communication DB control system 220.

Detailed Description Paragraph Right (193):

(2) A process that accesses the database system 240 corresponding to the message data and that references and updates data of files of the database 244 in the database system 240 (these files includes a customer master file 2441, a product master file 2442, an inventory master file 2443, and a transaction history file 2444). In addition, a process that generates data (reply data) registered to the reply table 352 of the public service A of the communication DB control system 220 according to the definition of the reply table definition table 303R of the public service A. This reply data is a reply result of a request of the client application 110 to the database server 200. However, the server application does not need to consider the reply data. Instead, corresponding to the content of the replay table definition table 303R, the reply table can be generated.

Detailed Description Paragraph Right (195):

(4) A process that requests the message reply control portion 230 in the communication DB control system 220 to inform the client application 110 of the completion of the execution of the SQL statement (INSERT) with the RETURN command.

Detailed Description Paragraph Right (196):

The client application 110 and the server application 210 that have been generated in the above-described manner send and receive data through the communication DB control system 220. In this case, the client application 110 accesses the communication DB control system 220 of the database server 200 only twice. In other words, when a request is issued, the input table 351 of the public service A is accessed one time. When the reply result of the request is received, the reply table 352 of the public service A is accessed one time. The server application 210 performs a back end process that searches and updates the database 244 of the database system 240. The server application 210 accesses the communication DB control system 220 only twice (the input table 351 and the reply table 352 are accessed one time each).

Detailed Description Paragraph Right (197):

The above-described process is an example of the reply type (real communication) data process. FIG. 15 is a schematic diagram showing a collecting type data process according to the embodiment that performs the order process that was performed by the client application 110 corresponding to the RDA.

Detailed Description Paragraph Right (198):

In this case, data items ("customer number", "part number", "number of orders", and "order number") that are supplied from the client application 110 to the server application 210 are defined in the input table definition information 303I of the



collecting type public service information 300. A process that receives data that is input by the user corresponding to a data item defined in the input table definition information 303I and that requests input data to be registered into the input table 351 of the communication DB control system 220 with the SQL statement (INSERT) is generated as the client application 110.

Detailed Description Paragraph Right (199):

On the other hand, with the SQL statement(SELECT), the database (DB) 244 of the database system 240 (that has a customer information file 2446, an inventory information file 2447, and an order information file 2448) is accessed by receiving input data of the input table 351 through the client application 110. Thus, performed are the back end processes as a server application 210 such as customer check, inventory check, inventory update, order registration, and so forth.

Detailed Description Paragraph Right (200):

Thus, when the client application 110 accesses the database server 200 one time (namely, sends the SQL statement (INSERT), which inputs the input data to the input table 351, to the database server 200), one order process is completed. Thus, the process is completed in a shorter time period than the conventional method. In addition, the traffic amount on the network is reduced and the communication cost is decreased.

Detailed Description Paragraph Right (201):

In the delivering type (delayed communication), the client application 110 and the server application 210 can be independently generated corresponding to the content of the definition of the input table definition information 303I of the communication DB 221.

Detailed Description Paragraph Right (202):

Since only the server application 210 accesses the database system 240, even if the construction of the database 244 of the database system 240 is changed, it is not necessary to correct the client application 110.

Detailed Description Paragraph Right (203):

FIG. 16 shows an example of the process of the database generated according to the public service of the communications DB control unit 220. In FIG. 16, the client application 110 instructs the server application 210 to update the data in the database 244 and the server application 210 informs the client application of the completion of the data update.

Detailed Description Paragraph Right (204):

FIGS. 17A and 17B are schematic diagrams showing a process in the case that even if the construction of the database 244 is changed, the correction of the client application 110 is not required. This process accords with the process shown in FIG. 1.

Detailed Description Paragraph Right (205):

As shown in FIG. 17A, the server application 210 receives request data from the client application 110 through the input table 351 (communication DB-TABLE1) in the communication DB control system 220 and stores the request data in the TABLE1 (DB1-TABLE1) of the database 244 (DB1) of the database system 240.

Detailed Description Paragraph Right (206):

In this case, the number of items of the DB1-TABLE1 increases and the access performance to the database DB1 deteriorates. As shown in FIG. 17B, in the database DB1, the DB1-TABLE1 is divided into two portions DB1-TABLE1-1 and DB1-TABLE1-2. In this case, only the server application 210 is corrected. In other words, one statement "INSERT DB1-TABLE1" in the server application 210 shown in FIG. 17A can be changed to "INSERT DB1-TABLE1-1" and "INSERT DB1-TABLE1-2" shown in FIG. 17B. This change does not affect the construction of the input table 351 (communication DB-TABLE1) of the communication DB control system 220. Thus, the change of the client application 110 is not required. Consequently, the manager of the database system 240 can change the structure of the database 244 without need to consider the client application 110.

Detailed Description Paragraph Right (207):

Only the public service information 300 (service definition information 301, input table definition information 303I, and reply table definition information 303R) registered to the communication DB 221 is open to the user (the developer of the



client application 110). The construction of the database 244 of the database system 240 of the database server 200 is not open to the user. Thus, the design and management of a table open to the user and the design and management of a table registered to the database 244 of the database system 240 can be separated. Thus, in designing a table of the database, the table of the communication DB 221 open to the user and the table of the database 244 of the database system 240 can be independently designed. Thus, the table designer (data provider) of the communication DB 221 can design and manage the table considering the operability for the user. The table designer (database manager) of the database 244 of the database system 240 can design and manage the table considering the accessibility and shared management of the database 244.

Detailed Description Paragraph Right (216):

First, the operation for registering the service definition information 301 open to the client application 110 to the communication DB 221 will be described. This registration operation can be performed by a user who has been assigned the update right of the above-described public service information 300.

Detailed Description Paragraph Right (221):

When the communication procedure control portion 201 receives a connection request from the client application 110 for the communication DB 221 through the network, the portion 201 sends a user identifier received along with the connection request to the user qualifying portion 202 and requests the user qualifying portion 202 to determine whether or not the executor of the client application 110 has the use right to the communication DB 221 (at step S1).

Detailed Description Paragraph Right (223):

The communication procedure control portion 201 receives the determined result. When the communication procedure control portion 201 has determined that the user has the use right of the communication DB 221 (YES at step S3), it sends a reply message that represents that "the connection to the communication DB 221 is successful" to the client application 110 (at step S4) and waits for an access request to the table of the communication DB 221 from the client application 110 (at step S5).

Detailed Description Paragraph Right (224):

When the user does not have the use right of the communication DB 221 (NO at step S3), the communication procedure control portion 201 sends a reply message that represents that "the connection to the communication DB 221 is unsuccessful" to the client application 110 (at step S6).

Detailed Description Paragraph Right (230):

When the SQL analyzing control portion 222C has determined that the client does not have a valid access right to the designated table (NO at step S13), the portion 222C informs the client (client application 110) of a reply message that represents "no access right" through the communication procedure control portion 201 (at step S15).

Detailed Description Paragraph Right (232):

The communication DB control system 220 is disposed between the client application 110 and the server application 210. The data storing process (communication DB storing process) by the client application 110 into the communication DB 221 is performed independently of the access (real DB access process) to the real DB 244 by the server application 210 using the stored data by the queue control system in the communication DB control system 220. These processes are asynchronously performed.

Detailed Description Paragraph Right (234):

While the communication DB control system 220 is waiting for a request for storing data to the input table 351 of the public service of the message queue control portion 225 in the communication DB control system 220, when the client application 110 issues such a request (INSERT) to the input table 351 (at step S31), the message queue control portion 225 queues the data to the input table 351. In other words, the data is stored in a communication buffer 226 formed in the memory and linked to the queue of the input table 351 (at step S32).

Detailed Description Paragraph Right (235):

After the queuing process has been completed, the message queue control portion 225 informs the requesting client application 110 of a reply message that represents that the storage request to the communication DB 221 has been completed (at step S33).

Detailed Description Paragraph Right (236):

The message queue control portion 225 determines the present state of the server application 210 that processes the data queued in the input table 351. While the server application 210 is processing the data, the flow returns to step S31. At step S31, the message queue control portion 225 waits until the client application 110 issues a data storing request ("processing" at step S34). On the other hand, while the server application 210 is waiting for a process ("waiting for process" at step S34), the message queue control portion 225 requests the application execution control portion 241 to activate the server application 210 through the application activation request control portion 229 (at step S35).

Detailed Description Paragraph Right (238):

The server application 210 retrieves data from the communication DB 221 (namely, the input table 351) stored by the client application 110, accesses the real DB 244 corresponding to the data, and performs a predetermined data process (at step S39).

Detailed Description Paragraph Right (241):

Corresponding to the data storing request from the client application 110 to the communication DB 221, when the data has been queued to the input table 351 of the public service, the communication DB control system 220 informs the client application 110 of the completion of the data storing request. Thus, the client application 110 can always have a quick reply to an access to the communication DB 221. In addition, the communication DB control system 220 can receive accesses from many client applications to the communication DB 221 in a short time.

Detailed Description Paragraph Right (256):

In FIG. 23, when the message queue control portion 225 receives a data storing request from the client application 110 (at step S51), the message queue control portion 225 determines the present storing service state of the input table 351 (at step S52). When the storing service is closed, the message queue control portion 225 informs the client application 110 of an error message that represents "data storage unsuccessful" through the communication procedure control portion 201 ("data storage invalid", at step S52).

Detailed Description Paragraph Right (258):

On the other hand, unless the storing service is closed at the step S52 ("data storage valid", at step S52), the required stored data is queued in the input table 351 (at step S53). When the data has been queued, the message queue control portion 225 informs the client application 110 of a reply message that represents that data storing request has been completed (at step S54).

Detailed Description Paragraph Right (260):

In other words, while the executing environment of the server application 210 stops, the access of the server application 210 to the real DB 244 corresponding to a service request from the client application 110 is stopped.

Detailed Description Paragraph Right (263):

As shown in FIG. 8, in this embodiment, the communication DB control system 220 contains two code converting/attribute changing portions 223C and 223S. In the communication DB 221, input tables 351 and 353 and a reply table 352 that are used to exchange a message (data) between the client application 110 and the server application 210 are generated. The formats (attributes) of the items of data stored in these tables that function as queues are defined as item attributes in table definition information 303 (input table definition information 303I and reply table definition information 303R) registered in the communication DB 221.

Detailed Description Paragraph Right (264):

FIG. 24 is a schematic diagram showing an item attribute converting system for data exchanged between the client application 110 and the server application 210 according to the embodiment.

Detailed Description Paragraph Right (265):

The code converting/attribute converting portion 223C performs the following attribute converting process for each item of data that the client application 110 sends and receives.

Detailed Description Paragraph Right (273):

When the code converting/attribute converting portion 223C receives data stored in

the communication DB 221 (input table 351) requested by the client application 110 from the SQL analyzing control portion 222C (at step S111), the portion 223C compares the item attribute information defined in the definition information 303I of the input table 351 of the communication DB 221 and the attribute of each item of the received data (at step S112). When there is an item in which attributes do not match ("no match", at step S112), the code converting/attribute converting portion 223C converts the attribute of the item into an attribute defined in the input table definition information 303I (at step S113).

Detailed Description Paragraph Right (276):

When data that is stored in the input table 351 by the client application 110 and that is sent to the server application 210 is retrieved from the input table 351 (at step S121), the code converting/attribute converting portion 223S references the definition information 303I of the input table 351 and compares an attribute of each item of the retrieved data and an attribute requested by the server application 210 (at step S122).

Detailed Description Paragraph Right (279):

When the client application 110 sends data to the server application 210 corresponding to the attribute converting process shown in FIGS. 25 and 26, the applications 110 and 210 process data with their proper attributes.

Detailed Description Paragraph Right (285):

When data to be sent to the client application (data supplied from the server application 210) is retrieved from the communication DB 221 (reply table 352 or input table 353) (at step S141), the code converting/attribute converting portion 223C references the definition information (303R or 303I) of the reply table 352 or input table 353 and determines whether or not attributes of all items of the retrieved data match attributes requested by the client application 110 (at step S142).

Detailed Description Paragraph Right (286):

When there is an item whose attributes do not match ("no match", at step S142), the code converting/attribute converting portion 223C converts the item into an attribute requested by the client application 110 (at step S143). The converted data is sent to the client application 110 through the communication procedure control portion 201 (at step S144).

Detailed Description Paragraph Right (287):

When the attributes of all the items of the data retrieved at the step S142 match the attributes requested by the client application 110 ("match", at step S142), the step S144 is immediately executed.

Detailed Description Paragraph Right (288):

When data is sent from the server application 210 to the client application 110 corresponding to the item attribute converting processes shown in FIGS. 27 and 28, the applications 210 and 110 can process data with their proper attributes.

Detailed Description Paragraph Right (289):

In this embodiment, attributes of data items to be exchanged between the client application 110 and the server application 210 are not defined. Thus, when numeric values are exchanged, even if attributes of an application of the client machine 100 do not match those of the database server 200, the numeric values can be exchanged. For example, data can be exchanged between a server application 210 using the DECIMAL attribute as a numeric attribute and a client application 110 with the INTEGER attribute as a numeral attribute.

Detailed Description Paragraph Right (290):

Thus, the client application 110 and the server application 210 can designate attributes of items of data corresponding to their process types.

Detailed Description Paragraph Right (292):

In this embodiment, the executing environment is present for each communication DB 221. This is because the service definition information 301 of the communication DB 221 correlates the public services with the server application 210 that executes the data process thereof. When the client application 110 uses (accesses) the communication DB 221, to start it, a CONNECT command is declared. To stop it, a DISCONNECT command is declared. When the CONNECT command is issued, the communication DB 221 is allocated the client application 110.

Detailed Description Paragraph Right (295):

FIG. 30 is a schematic diagram showing a process of which a client application 110-1 (#1) performs a job using the public service A and a client application 110-2 (#2) performs a job using the public services A and B. In this case, the client application 110-1 (#1) and the client application 1102 (#2) can communicate with the server application A.

Detailed Description Paragraph Right (296):

In this embodiment, the communication DB control system 220 can independently perform the operation tests of the client application 110 and the server application 210.

Detailed Description Paragraph Right (298):

(1) The client application 110 writes data to the input table 351 of the communication DB control system 220 corresponding to an SQL programming (INSERT).

Detailed Description Paragraph Right (299):

(2) After the client application 110 has written the data to the input table 351, the application 110 determines the content thereof.

Detailed Description Paragraph Right (302):

In this embodiment, since the communication DB control system 200 is accessed in the same method as the issuance of a conventional SQL programming, commercially available spreadsheet software and database access (database access) software can be used as a client application 110.

Detailed Description Paragraph Right (304):

The client application and the server application can exchange data (messages) by storing and retrieving data to and from a table with a format defined corresponding to public service information (this table is referred to as a public table). Thus, the client application and the server application can be independently developed by considering only the public service information.

Detailed Description Paragraph Right (306):

The client application can be generated by considering only an access to the public table rather than the format of a table formed in the real database (this table is referred to as a real table). The server application accesses the database corresponding to the data retrieved from the public table.

Detailed Description Paragraph Right (307):

Thus, the construction of the databases can be changed by correcting only the server application rather than the client application.

Detailed Description Paragraph Right (308):

Thus, the manager of the database can change the construction of the real database without considering the client application.

Detailed Description Paragraph Right (311):

Since the server application can access a plurality of real tables in the real database with one public table of the client application, the traffic amount on the network between the client application and the server application can be reduced and the job process can be performed at high speed.

Detailed Description Paragraph Right (313):

Since the functional logic for the data process type such as real communication/delayed communication is omitted from the application (client application and server application) side, the interface between the client application and the server application is only the data format. Thus, both the applications can be independently constructed. Thus, the productivity of both the applications is improved.

Detailed Description Paragraph Right (314):

Since the data attribute converting process logic is omitted from both the client application and the server application, the productivity of both the applications is improved.

Detailed Description Paragraph Right (315):

When the present invention is used for an alternative of a driver for the server

application and for an alternative of a stub for the client application, their operation tests can be independently performed without need to provide both the applications.

Detailed Description Paragraph Right (316):

In addition, commercial application packages such as existing spreadsheet software and database access software can be easily used as a client application.

Detailed Description Paragraph Left (10):

Operation test of client application 110

Detailed Description Paragraph Type 1 (3):

(3) Name of server application 210 to which data (message) being input by client application 110 is sent, and

Detailed Description Paragraph Type 1 (22):

f) The SQL analyzing control portion 222C informs the client application 110 of the completion of the registration through the communication procedure control portion 201.

Detailed Description Paragraph Type 1 (24):

Each item of data received from the client application 110 is converted into an attribute defined by table definition information and then output.

Detailed Description Paragraph Type 1 (25):

Each item of data (stored in the reply table 352 or the input table 353) stored in the communication DB 221 by the server application 210 is converted into an item attribute required by the client application 110 and then output.

CLAIMS:

1. A communication control apparatus provided between a client application and a server application and controlling transmission of data between the client application and the server application in a client/server system, said apparatus comprising:

a first database storing published service information being published for a client and storing according to the published service information data transmitted between the client application and the server application, said published service information comprising information defining a format of stored data; and

control means for controlling data communications between the client application and the server application by temporarily storing in said first database the data received from one of the client application and the server application, wherein an interface to the first database available to the client application is independent of an interface between the first database and the server application, and wherein the client application retrieves the published service information from the first database.

5. A communication control apparatus provided between a client application and a server application and controlling transmission of data between the client application and the server application in a client/server system, said apparatus comprising:

a first database storing published service information being published for a client and storing according to the published service information data transmitted between the client application and the server application;

client request analyzing means for analyzing a request received from the client application and storing data specified by a write request in said first database; and

server request analyzing means for analyzing a request received from the server application, reading data specified by a read request from said first database, and transmitting the data to the server application, wherein an interface to the first database available to the client application is independent of an interface between the first database and the server application, and wherein the client application retrieves the published service information from the first database.

6. A communication control apparatus for controlling transmission of data between a client application and a server application in a client/server system, comprising:

a first database used in data communications between the client application and the server application;

client request analyzing means for analyzing a request received from the client application and storing data specified by a write request in said first database;

server request analyzing means for analyzing a request received from the server application, reading data specified by a read request from said first database, and transmitting the data to the server application;

first entry means for entering in said first database first security information about a client authorized to access said first database; and

first security check means for referring to the first security information when a request for connection to said first database is made from a client through the client application, and for determining whether the request of the client to access said first database is accepted.

8. The communication control apparatus according to claim 7, wherein

said first database comprises an entry table in which data is written during a write request from the client application and whose format is defined according to the published service information.

9. A communication control apparatus for controlling transmission of data between a client application and a server application in a client/server system, comprising:

a first database used in data communications between the client application and the server application, said first data base comprising:

first entry means for entering published service information defining information about said first database published to clients, and

an entry table in which data is written at a write request from the client application and whose format is defined according to the published service information;

client request analyzing means for analyzing a request received from the client application and storing data specified by a write request in said first database;

server request analyzing means for analyzing a request received from the server application, reading data specified by a read request from said first database, and transmitting the data to the server application;

second entry means for entering in said first database second security information about a client authorized to access said entry table; and

second security check means for referring to the second security information when a request for connection to said entry table is made from a client through the client application, and for determining whether the request of the client to access said entry table is accepted.

11. The communication control apparatus according to claim 10, wherein

said client request analyzing means comprises first queue control means for storing data received from the client application queue in said entry table.

12. A communication control apparatus for controlling transmission of data between a client application and a server application in a client/server system, comprising:

a first database used in data communications between the client application and the server application, said first data base comprising:

first entry means for entering published service information defining information about said first database published to clients, and

an entry table in which data is written at a write request from the client application and whose format is defined according to the published service information, said entry table referring to a queue;

client request analyzing means for analyzing a request received from the client application and storing data specified by a write request in said first database, said client request analyzing means comprising first queue control means for having data received from the client application queue in said entry table; and

server request analyzing means for analyzing a request received from the server application, reading data specified by a read request from said first database, and transmitting the data to the server application, wherein

said first queue control means prevents data requested by the client application from being queued on said entry table when a block instruction is issued for a service provided according to the published service information in which the format of said entry table is defined.

13. A communication control apparatus for controlling transmission of data between a client application and a server application in a client/server system, comprising:

a first database used in data communications between the client application and the server application, said first data base comprising:

first entry means for entering published service information defining information about said first database published to clients, and

an entry table in which data is written at a write request from the client application and whose format is defined according to the published service information, said entry table referring to a queue;

client request analyzing means for analyzing a request received from the client application and storing data specified by a write request in said first database, said client request analyzing means comprising first queue control means for having data received from the client application queue in said entry table; and

server request analyzing means for analyzing a request received from the server application, reading data specified by a read request from said first database, and transmitting the data to the server application, wherein

said client request analyzing means further comprises notifying means for notifying the client application of completion of a process requested by the client application when the data received from the client application are completely queued on said entry table by said first queue control means.

20. The communication control apparatus to claim 8, wherein

said first database comprises a response table to which response data addressed to the client application is written during a write request of the server application and in which a format is defined according to the published service information.

24. The communication control apparatus to claim 23, further comprising:

third attribute converting means for converting an attribute of each item of the data retrieved from said response table into an appropriate attribute for a process performed by the client application by referring to attribute information of each item of the data defined by the published service information, wherein

said client request analyzing means transmits the data whose attribute is converted by said third attribute converting means to the client application.

25. A communication control apparatus provided between a client application and a server application and controlling transmission of data between the client application and the server application in a client/server system, comprising:

a first database storing published service information published for a client and storing according to the published service information data transmitted between the client application and the server application;

server request analyzing means for analyzing a request received from the server application and storing data specified by a write request in said first database; and

client request analyzing means for analyzing a request received from the client application, reading data specified by a read request from said first database, and transmitting the data to the client application, wherein an interface to the first database available to the client application is independent of an interface between the first database and the server application, and wherein the client application retrieves the published service information from the first database.

31. The communication control apparatus to claim 30, further comprising:

second attribute converting means for converting an attribute of each item of the data received by said entry table by said client request analyzing means into an appropriate attribute for a process performed by the client application by referring to an attribute information about each item of the data defined by the published service information, wherein

said client request analyzing means transmits the data whose attribute is converted by said second attribute converting means to the client application.

32. A communication control apparatus for controlling transmission of data between a client application and a server application in a client/server system, comprising:

a first database used in data communications between the client application and the server application;

server request analyzing means for analyzing a request received from the server application and storing data specified by a write request in said first database; and

client request analyzing means for analyzing a request received from the client application, reading data specified by a read request from said first database, and transmitting the data to the client application, wherein

said server application accesses a second database; and

a record on one table in said first table comprises at least one item belonging to each record on at least one table in said second database.

33. A client/server system comprising a client application and a server application which performs a job in cooperation with the client application, said apparatus comprising:

a database storing published service information being published for a client and storing according to the published service information data transmitted between the client application and the server application;

client request analyzing means for receiving data addressed from the client application to the server application and for entering the data in said database; and

server request analyzing means for retrieving from the database the data addressed to the server application and transmitting the data to the server application, wherein an interface to the database available to the client application is independent of an interface between the database and the server application, and wherein the client application retrieves the published service information from the database.

36. A client/server system comprising a client application and a server application which accesses a first database in cooperation with the client application, comprising:

a second database storing published service information published for a client and storing according to the published service information data transmitted between the client application and the server application;

client request analyzing means for receiving data addressed from the client



application to the server application and for entering the data in said second database; and

server request analyzing means for retrieving from the second database the data addressed to the server application and transmitting the data to the server application, wherein

said server application accesses the first database based on data received from said server request analyzing means, wherein an interface to the second database available to the client application is independent of an interface between the second database and the server application, and wherein the client application retrieves the published service information from the second database.

37. A client/server system comprising a client application and a server application which accesses a first database in cooperation with the client application, comprising:

a second database storing data transmitted between the client application and the server application;

client request analyzing means for receiving data addressed from the client application to the server application and entering the data in said second database; and

server request analyzing means for retrieving from the database the data addressed to the server application and transmitting the data to the server application, wherein

said server application accesses the first database based on data received from said server request analyzing means, and wherein an item belonging to a record in the first database is located as being distributed to records on a plurality of tables in said second database.

40. The client/server system according to claim 38, further comprising:

control means for restricting access of the client application, which accesses the first database through said second database, to said second database depending on the load of a database server for managing the first database.

42. A client/server system comprising a client application and a server application which accesses a first database in cooperation with the client application, comprising:

a second database storing data transmitted between the client application and the server application, said second database comprising published service information indicating services published to clients and defining a format of a table of the service;

client request analyzing means for receiving data addressed from the client application to the server application and for entering the data in said second database;

server request analyzing means for retrieving from the database the data addressed to the server application and transmitting the data to the server application, wherein said server application accesses the first database based on data received from said server request analyzing means;

control means for restricting access to the first database of the server application assigned to each of the services published to the clients depending on an importance level set for each of the services published to the clients, wherein said server application is provided individually for each of the services published to the clients.

43. A client/server system comprising a client application and a server application which accesses a first database in cooperation with the client application, comprising:

a second database storing data transmitted between the client application and the server application, said second database comprising published service information

indicating services published to clients and defining a format of a table of the service;

client request analyzing means for receiving data addressed from the client application to the server application and for entering the data in said second database;

server request analyzing means for retrieving from the database the data addressed to the server application and transmitting the data to the server application, wherein said server application accesses the first database based on data received from said server request analyzing means; and

control means for restricting access of the client application, which accesses said second database provided by the service published to the clients, to said second database depending on an importance level of the service published to the clients.

44. A method of accessing a database of a client/server computer system, comprising the steps of:

writing by a client application data addressed to a server application to a first database, said first database storing published service information published for a client and storing according to the published service information the data, said published service information comprising information defining a format of stored data; and

retrieving by said server application data requested by the client application from the first database and accessing a second database based on retrieved data, wherein an interface to the first database available to the client application is independent of an interface between the first database and the server application, and wherein the client application retrieves the published service information from the first database.

45. A database processing method operated in a database system of a client/server computer system, comprising the steps of:

writing by a client application data addressed to a server application to a first database, said first database storing published service information published for a client and storing according to the published service information the data;

retrieving by said server application data requested by the client application from the first database, accessing a second database according to retrieved data, generating response data addressed to the client application, and writing the generated data to the first database; and

reading by the client application the response data from the first database, wherein an interface to the first database available to the client application is independent of an interface between the first database and the server application, and wherein the client application retrieves the published service information from the first database.

46. A method of transmitting a message between a client application and a server application in a client/server computer system, comprising the steps of:

writing by said client application to a database the message addressed to the server application, said database storing published service information published for a client and storing according to the published service information the message; and

reading by said server application the message from the database, wherein an interface to the database available to the client application is independent of an interface between the database and the server application, and wherein the client application retrieves the published service information from the database.

47. A method of transmitting a message between a client application and a server application in a client/server computer system, comprising the steps of:

writing by said server application to a database a message addressed to the client application, said database storing published service information published for a client and storing according to the published service information the message; and

reading by said client application the message from the database, wherein an interface to the database available to the client application is independent of an interface between the database and the server application, and wherein the client application retrieves the published service information from the database.

**WEST**

Generate Collection

Print

L1: Entry 471 of 3066

File: USPT

Feb 12, 2002

DOCUMENT-IDENTIFIER: US 6347398 B1

TITLE: Automatic software downloading from a computer network

Brief Summary Paragraph Right (2):

The Internet is a world-wide network of cooperating computer networks. Connected to the Internet are thousands of individual computers, each with a variety of application programs. From a user's point of view, access to the Internet and its services typically are accomplished by invoking a network application program (e.g., a network browser). The network application program acts as an interface between the user and the Internet. Network application programs are typically "client" applications that accept commands from the user and obtain Internet data and services by sending requests to "server" applications on other computers at other locations on the Internet.

Detailed Description Paragraph Right (14):

The Component Object Model (COM) and Distributed Component Object Model (DCOM) are models used for object oriented programming. The COM and DCOM specifies how objects within a single application or between applications (e.g. client/server applications) interact and communicate by defining a set of standard interfaces. Interfaces are groupings of semantically related functions through which a client application accesses the services of a server application.

Detailed Description Paragraph Right (19):

The browser and other client applications within the local computer 34 preferably work with documents (which have data formats other than those native to the browser or client application) by encapsulating the document's data into an associated object, and integrating with the object using pre-defined interfaces as described more fully below. This allows full use of all the features described below.

Detailed Description Paragraph Right (20):

Alternatively, the browser or other client application can work with a document by launching an application program associated with the document and causing the associated application program to load the document and perform a specified operation on the document. In the Microsoft Windows.RTM. 95 operating system, this is done by looking up the application program associated with the document's file name extension in the system registry, (e.g., ".doc" for Microsoft.RTM. Word documents, ".vsd" for Shapeware's VISIO drawings, etc.) and a text string command for directing the program to perform a desired operation. The application program associated with a document also may be identified in the Microsoft Windows.RTM. operating system by matching a bit pattern at specific offset into the file as also specified in the system registry. The browser or other client application patches the document's file name into the text string command and invokes the operating system's shell execute service to execute the command. Shell execute is a well known Windows.RTM. operating system service which issues a text string as a command to the operating system's command interpreter (i.e., the "command.com" program in the Windows.RTM. operating system).

Detailed Description Paragraph Right (34):

The CLASSID attribute is used to specify an object identifier. On each computer (36,44) connected to the computer network 38, for example, the operating system registry (also called the registration database) is used to store relevant information about object oriented components according to their CLaSS Identifier (CLSID). An object registers its CLSID in the operating system registry database to enable client applications (e.g., a network browser) to locate and load the executable code associated with the objects. The HTML <OBJECT> CLASSID is equivalent

to the object-oriented CLSID.

Detailed Description Paragraph Right (71):

A code install operation 74 requires additional services from the client in order to complete the negotiation necessary for a download operation 70. Such services are requested using IBindStatusCallback::QueryInterface. The specific interface requested in IBindStatusCallback::QueryInterface is ICodeInstall. This interface is implemented by a client application (e.g., a network browser). The ICodeInstall interface has two member functions: NeedVerificationUI and OnCodeInstallProblem. The ICodeInstall interface and functions are shown below.

Detailed Description Paragraph Right (93):

When all of the desired computer software components have been located 68, downloaded 70, verified 72, installed 74, and registered 76, any appropriate software components are returned 78 to the client application (e.g., the network browser) which made the request for the computer software components. The software components returned to the client application are then displayed in the document display area 52 on the display device 20 of the local computer 36.

Detailed Description Paragraph Right (96):

However, other applications which are not network browsers, and do not parse HTML documents with <OBJECT> or <EMBED> tags can also use system 56 and methods 66 to download any type of computer software component from a local or remote computer by passing appropriate parameters to CoGetClassObjectFromURL interface. The computer software components need not provide dynamic or interactive multimedia to a client application. In addition interfaces other than CoGetClassObjectFromURL could also be used.

**WEST****Freeform Search****Database:**

US Patents Full-Text Database  
US Pre-Grant Publication Full-Text Database  
JPO Abstracts Database  
EPO Abstracts Database  
Derwent World Patents Index  
IBM Technical Disclosure Bulletins

**Term:**

L22 and (primary near stor\$ and secondary stor\$)

**Display:**  **Documents in Display Format:**  **Starting with Number** **Generate:** ☐ Hit List ☒ Hit Count ☐ Side by Side ☐ Image

---

**Search History**

---

**DATE:** Thursday, March 07, 2002    [Printable Copy](#)    [Create Case](#)

| <u>Set Name</u><br>side by side                          | <u>Query</u>                                       | <u>Hit Count</u> | <u>Set Name</u><br>result set |
|--|--|------------------|-------------------------------|
| <i>DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR</i> |  |                  |                               |
| <u>L25</u>   | L24 and cach\$ near5 subsystem                     | 7                | <u>L25</u>                    |
| <u>L24</u>   | L22 and (primary near stor\$ and secondary stor\$) | 444              | <u>L24</u>                    |
| <u>L23</u>   | L22 and primary and secondary stor\$               | 1438992          | <u>L23</u>                    |
| <u>L22</u>   | L21 and database near5 manager                     | 505              | <u>L22</u>                    |
| <u>L21</u>   | client near5 application                           | 7409             | <u>L21</u>                    |
| <u>L20</u>   | L19 and control near program                       | 15               | <u>L20</u>                    |
| <u>L19</u>   | L17 and data near record near5 stor\$              | 188              | <u>L19</u>                    |
| <u>L18</u>   | L17 and data near record                           | 1714             | <u>L18</u>                    |
| <u>L17</u>   | database near system                               | 10624            | <u>L17</u>                    |
| <u>L16</u>   | L15 and database                                   | 2                | <u>L16</u>                    |
| <u>L15</u>   | trash near2 manag\$ near2 system                   | 18               | <u>L15</u>                    |
| <u>L14</u>   | L13 and ((707/206 )!.CCLS. )                       | 18               | <u>L14</u>                    |
| <u>L13</u>   | garbage near manag\$                               | 49               | <u>L13</u>                    |
| <u>L12</u>   | 5706506.pn.  | 3                | <u>L12</u>                    |
| <u>L11</u>   | 5717893.pn.  | 3                | <u>L11</u>                    |
| <u>L10</u>   | L8 and cach\$ same system                          | 56               | <u>L10</u>                    |
| <u>L9</u>  | L8 and divide near sections                        | 0                | <u>L9</u>                     |
| <u>L8</u>  | L7 and database near structure                     | 195              | <u>L8</u>                     |
| <u>L7</u>  | L6 and stor\$ and retriev\$ near records           | 663              | <u>L7</u>                     |
| <u>L6</u>  | database near5 manag\$                             | 14731            | <u>L6</u>                     |
| <u>L5</u>  | L4 and client near application                     | 1                | <u>L5</u>                     |
| <u>L4</u>  | L3 and control near program                        | 100              | <u>L4</u>                     |
| <u>L3</u>  | L2 and pages                                       | 462              | <u>L3</u>                     |
| <u>L2</u>  | L1 and (secondary near storage)                    | 1410             | <u>L2</u>                     |
| <u>L1</u>  | (primary near storage or main near storage)        | 17722            | <u>L1</u>                     |

END OF SEARCH HISTORY

**WEST**[Help](#)[Logout](#)[Interrupt](#)[Main Menu](#)[Search Form](#)[Posting Counts](#)[Show S Numbers](#)[Edit S Numbers](#)[Preferences](#)[Cases](#)**Search Results -**

| Terms                | Documents |
|----------------------|-----------|
| cache near subsystem | 591       |

**Database:**

US Patents Full-Text Database  
US Pre-Grant Publication Full-Text Database  
JPO Abstracts Database  
EPO Abstracts Database  
Derwent World Patents Index  
IBM Technical Disclosure Bulletins

**Search:**[Refine Search](#)[Recall Text](#)[Clear](#)**Search History****DATE:** Thursday, March 07, 2002   [Printable Copy](#)   [Create Case](#)



| <u>Set Name</u><br>side by side                          | <u>Query</u>                    | <u>Hit Count</u> | <u>Set Name</u><br>result set |
|--|---------------------------------|------------------|-------------------------------|
| <i>DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR</i> |                                 |                  |                               |
| <u>L10</u>   | cache near subsystem            | 591              | <u>L10</u>                    |
| <u>L9</u>  | l1 and cache near subsystem     | 2                | <u>L9</u>                     |
| <u>L8</u>  | L3 and cache near subsystem     | 0                | <u>L8</u>                     |
| <u>L7</u>  | L4 and cache same subsystem     | 0                | <u>L7</u>                     |
| <u>L6</u>  | L4 and cache near subsystem     | 0                | <u>L6</u>                     |
| <u>L5</u>  | L4 and cache near5 subsystem    | 0                | <u>L5</u>                     |
| <u>L4</u>  | L3 and client near2 application | 53               | <u>L4</u>                     |
| <u>L3</u>  | L2 and control near2 program    | 387              | <u>L3</u>                     |
| <u>L2</u>  | L1 and data near2 records       | 1637             | <u>L2</u>                     |
| <u>L1</u>  | database adj manag\$            | 7758             | <u>L1</u>                     |

END OF SEARCH HISTORY

**WEST**

Generate Collection

Print

L7: Entry 7 of 27

File: USPT

Oct 14, 1997

US-PAT-NO: 5678042

DOCUMENT-IDENTIFIER: US 5678042 A

TITLE: Network management system having historical virtual catalog snapshots for  
overview of historical changes to files distributively stored across network domain

DATE-ISSUED: October 14, 1997

## INVENTOR-INFORMATION:

| NAME             | CITY        | STATE | ZIP CODE | COUNTRY |
|------------------|-------------|-------|----------|---------|
| Pisello; Thomas  | De Bary     | FL    |          |         |
| Crossmier; David | Casselberry | FL    |          |         |
| Ashton; Paul     | Oviedo      | FL    |          |         |

## ASSIGNEE-INFORMATION:

| NAME                     | CITY          | STATE | ZIP CODE | COUNTRY | TYPE CODE |
|--------------------------|---------------|-------|----------|---------|-----------|
| Seagate Technology, Inc. | Scotts Valley | CA    |          |         | 02        |

APPL-NO: 8/ 590528 [PALM]

DATE FILED: January 24, 1996

## PARENT-CASE:

This application is a division of Ser. No. 08/153,011, filed Nov. 15, 1993, now U.S.  
Pat. No. 5,495,607.

INT-CL: [6] G06 F 7/00, G06 F 7/06, G06 F 12/00, G06 F 17/30

US-CL-ISSUED: 395/610; 395/607, 395/200.03, 395/200.01, 395/800, 395/650, 395/280

US-CL-CURRENT: 714/47; 707/10, 707/7, 709/224, 710/100FIELD-OF-SEARCH: 395/600, 395/200.03, 395/200.11, 395/200.13, 395/800, 395/440,  
395/444, 395/200.06, 395/200.12, 395/200.17, 395/200.18, 395/182.02

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected

Search ALL

|                          | PAT-NO         | ISSUE-DATE     | PATENTEE-NAME       | US-CL     |
|--------------------------|----------------|----------------|---------------------|-----------|
| <input type="checkbox"/> | <u>4141006</u> | February 1979  | Braxton et al.      | 340/505   |
| <input type="checkbox"/> | <u>4710870</u> | December 1987  | Blackwell et al.    | 364/200   |
| <input type="checkbox"/> | <u>4805134</u> | February 1989  | Calo et al.         | 395/600   |
| <input type="checkbox"/> | <u>4897841</u> | January 1990   | Gang, Jr.           | 370/85.13 |
| <input type="checkbox"/> | <u>4914571</u> | April 1990     | Baratz et al.       | 395/600   |
| <input type="checkbox"/> | <u>4987531</u> | January 1991   | Nishikado et al.    | 395/600   |
| <input type="checkbox"/> | <u>5001628</u> | March 1991     | Johnson et al.      | 395/600   |
| <input type="checkbox"/> | <u>5077658</u> | December 1991  | Bendert et al.      | 395/600   |
| <input type="checkbox"/> | <u>5133075</u> | July 1992      | Risch               | 395/800   |
| <input type="checkbox"/> | <u>5163131</u> | November 1992  | Row et al.          | 395/200   |
| <input type="checkbox"/> | <u>5175852</u> | December 1992  | Johnson et al.      | 395/600   |
| <input type="checkbox"/> | <u>5216591</u> | June 1993      | Nemirovsky et al.   | 395/200   |
| <input type="checkbox"/> | <u>5220562</u> | June 1993      | Takada et al.       | 370/85.13 |
| <input type="checkbox"/> | <u>5247670</u> | September 1993 | Matsunaga           | 395/650   |
| <input type="checkbox"/> | <u>5271007</u> | December 1993  | Kurahashi et al.    | 395/600   |
| <input type="checkbox"/> | <u>5287453</u> | February 1994  | Roberts             | 395/200   |
| <input type="checkbox"/> | <u>5287461</u> | February 1994  | Moore               | 305/275   |
| <input type="checkbox"/> | <u>5295244</u> | March 1994     | Dev et al.          | 395/200   |
| <input type="checkbox"/> | <u>5325527</u> | June 1994      | Cwikowski et al.    | 395/650   |
| <input type="checkbox"/> | <u>5403639</u> | April 1995     | Belsan et al.       | 395/600   |
| <input type="checkbox"/> | <u>5448727</u> | September 1995 | Annevelink          | 395/600   |
| <input type="checkbox"/> | <u>5506986</u> | April 1996     | Healy               | 395/600   |
| <input type="checkbox"/> | <u>5537585</u> | July 1996      | Blickenstaff et al. | 395/600   |

ART-UNIT: 235

PRIMARY-EXAMINER: Harvey; Jack B.

ASSISTANT-EXAMINER: Phan; Raymond N.

ATTY-AGENT-FIRM: Fliesler, Dubb, Meyer &amp; Lovejoy

## ABSTRACT:

A network management system includes a domain administrating server (DAS) that stores a virtual catalog representing an overview of all files distributively stored across a network domain currently or in the past. The current and historical file information is used for assisting in auditing or locating files located anywhere in the domain. The current file information is used for assisting in transferring files across the domain. The domain administrating server (DAS) also includes a rule-base driven artificial administrator for monitoring and reacting to domain-wide alert reports and for detecting problematic trends in domain-wide performance based on information collected from the network domain.

4 Claims, 8 Drawing figures

**WEST**[Help](#)[Logout](#)[Interrupt](#)[Main Menu](#)[Search Form](#)[Posting Counts](#)[Show S Numbers](#)[Edit S Numbers](#)[Preferences](#)[Cases](#)**Search Results -**

| Terms                          | Documents |
|--------------------------------|-----------|
| L7 and client same application | 3         |

**Database:**

US Patents Full-Text Database  
 US Pre-Grant Publication Full-Text Database  
 JPO Abstracts Database  
 EPO Abstracts Database  
 Derwent World Patents Index  
 IBM Technical Disclosure Bulletins

**Search:**

[Refine Search](#)[Recall Text](#)[Clear](#)**Search History**
**DATE: Thursday, March 07, 2002**
[Printable Copy](#)
[Create Case](#)

**Set Name**   **Query**  
 side by side

**Hit Count**   **Set Name**  
 result set

*DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR*

|           |  |       |           |
|-----------|--|-------|-----------|
| <u>L8</u> | L7 and client same application                   | 3     | <u>L8</u> |
| <u>L7</u> | L6 and control near program                      | 27    | <u>L7</u> |
| <u>L6</u> | L5 and sections                                  | 83    | <u>L6</u> |
| <u>L5</u> | L4 and page\$                                    | 139   | <u>L5</u> |
| <u>L4</u> | L3 and second\$ near stor\$                      | 235   | <u>L4</u> |
| <u>L3</u> | L2 and (primary near stor\$ or main near stor\$) | 541   | <u>L3</u> |
| <u>L2</u> | L1 and records                                   | 20128 | <u>L2</u> |
| <u>L1</u> | database same system                             | 51863 | <u>L1</u> |

END OF SEARCH HISTORY

**WEST**

Generate Collection

Print

L1: Entry 466 of 3066

File: USPT

Feb 19, 2002

DOCUMENT-IDENTIFIER: US 6349301 B1

TITLE: Virtual environment bystander updating in client server architecture

Detailed Description Paragraph Right (10):

Typically, a client application that incorporates the present invention would be loaded in nonvolatile storage 204, i.e., onto the hard drive of computer 200, from a removable floppy disk 214 that is inserted into floppy drive 212, or from removable CD-ROM 205. When the program comprising these machine instructions is to be executed, the machine instructions are loaded into memory 206, which includes both read only memory (ROM) and random access memory (RAM) (not separately shown). The machine instructions, once loaded into RAM, can be executed by CPU 202. The user provides input and controls the application being executed using the keyboard and/or mouse or other pointing device. Although the exemplary operating environment described herein employs a hard disk, floppy disk 214, and CD-ROM 205, it should be appreciated by those skilled in the art that other types of computer readable media, which can store data that are accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, RAMs, ROMs, and the like, may also be used for conveying machine instructions that define the present invention and which are used to define an operating environment for it.